

Title	地球シミュレータ上での18テラフロップス級及び1590億次元行列の厳密対角化計算:トラップされた強相関フェルミ原子ガスの基底状態探索(計算科学の基盤技術とその発展)
Author(s)	今村, 俊幸; 山田, 進; 町田, 昌彦
Citation	数理解析研究所講究録 (2006), 1505: 145-156
Issue Date	2006-07
URL	<a href="http://hdl.handle.net/2433/58510">http://hdl.handle.net/2433/58510</a>
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

地球シミュレータ上での18テラフロップス級及び  
1590億次元行列の厳密対角化計算：  
トラップされた強相関フェルミ原子ガスの基底状態探索

電気通信大学 今村 俊幸 (Toshiyuki Imamura)

University of Electro-Communications

日本原子力研究開発機構 山田 進 (Susumu Yamada)

Japan Atomic Energy Agency

日本原子力研究開発機構 町田 昌彦 (Masahiko Machida)

Japan Atomic Energy Agency

## 1 はじめに

本研究の計算物理での舞台となる強相関フェルミ原子ガスは、それ自身が持つ物理的興味の他、ナノスケールの高温超伝導体の物性を理解する上で重要な役割を果たすものである。例えば、図1のような閉じ込めポテンシャルを考慮した系において超流動の前駆現象となるクーパペアが現れるという興味深い数値結果が得られる[1]。本系の量子物性を研究する上で有力な手段として i) 量子モンテカルロ法, ii) 密度行列繰り込み群法, iii) 厳密対角化法がある。各手法の詳細については本稿の主題から外れるために省略するが、それぞれの利点を持ち計算複雑さや計算精度も異なる点を注意しておく。

我々が本論文で取り扱う厳密対角化法は、強相関電子系の振る舞いを記述する代表モデルハバードモデルに近似を入れることなくハミルトニアン行列(式(1))を対角化し、基底状態(最小固有値と固有ベクトル)を求めるものである。系に一切の近似を与えないことから、精度の面で最も優れている方法であるとされているが、要求メモリサイズが膨大なものとなるため大規模系への応用は限られてきた経緯がある。近年の64ビットOSの出現や地球シミュレータ[3]級のスパコン開発の結果、テラバイトサイズの計算が可能となり、大規模系の計算機実験が可能となってきた。

$$H = -t \sum_{i,j,\sigma} (a_{j,\sigma}^\dagger a_{i,\sigma} + \text{H.C.}) + U \sum_i n_{i\uparrow} n_{i\downarrow} + \left(\frac{2}{N}\right)^2 V \sum_{i,\sigma} n_{i,\sigma} \left(i - \frac{N}{2}\right)^2 \quad (1)$$

大規模系の計算機実験を遂行する上で、計算物理が主体となりカバーする範囲以外に興味深い問題点と解決すべき技術項目が浮上している。

1) 計算科学(特に数値線形計算)から本問題に対する興味は、ハミルトニアン行列の性質が大規模疎行列であることと最小(もしくは最小から幾つかの)固有対の計算のみが要求されることを十分に考慮した効率的な計算手法の研究・開発にある。

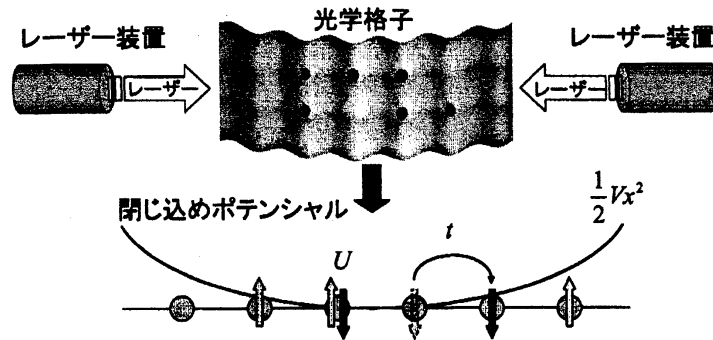


図 1: レーザ光の干渉によって作られる光学格子の概念図 (上) 及び光学格子上のフェルミ粒子の模式図 (下)

2) また、計算機科学の観点からは、本研究の遂行には地球シミュレータ級の大型並列計算機の利用が必須だが、従来のコード開発技術で効率的な処理を実現できるか実装面での問題解決が極めて重大な問題と認識される。

本報告は、この様に計算物理学、数値線形計算分野、計算機科学の最先端の技術融合によって成し得た、地球シミュレータ上での実効性能 18 テラフロップスおよび 1590 億次元の固有値計算の結果を報告する。特に、2005 年 11 月に行われた国際会議 Supercomputing2005(SC|05)での論文発表 [5] の以降に得られた最新結果も併せて紹介する。

## 2 ランチョス法と共役勾配法による大規模固有値計算

### 2.1 ランチョス法

ハバードモデルのハミルトニアン行列は式 (1) で表現され、対称疎行列であることと共に使用メモリへの要求から、基底状態 (最小固有値と対応する固有ベクトル) の計算にはランチョス法 (図 2 左) が伝統的に用いられてきた [2]。3 本のベクトルのみからなる反復形式の簡単さはあるものの、ランチョス法はクリロフ基底による十分な近似を保証するために予め反復回数を決定しなくてはならなかったり、(メモリの制約から) 反復過程で得られるランチョス基底を全て保持することが困難な場合には計算途中で固有ベクトルの計算時に冗長なランチョス反復を必要とするなど効率面での問題があった [4]。もちろん、複数固有対の計算を行う場合はランチョス基底の直交性保証の問題なども同時に考慮しなくてはならない。この様に安定性の改善において十分な注意を払わなくてはならない解法であることは数値解析分野の研究者には周知の事実であろう。

### 2.2 前処理付き共役勾配法

ランチョス法の問題点に対応するために、我々は共役勾配法の理論に基づいた反復解法を用いて、収束特性 (= 実行時間) をより改善する。本研究で用いるアルゴリズムは Knyazev が発表した方法 LOBPCG [6] に基づき、探索方向の更新にリッツベクトルから算出される

$x_0$ := an initial guess. $\beta_0 := 1, v_{-1} := 0, v_0 = x_0 / \ x_0\ $ do $i=0, 1, \dots, m-1$ , or until $\beta_i < \epsilon$ , $u_i := H v_i - \beta_i v_{i-1}$ $\alpha_i := (u_i, v_i)$ $w_i := u_i - \alpha_i v_i$ $\beta_i := \ w_i\ $ $v_{i+1} := w_i / \beta_{i+1}$ enddo	$x_0$ := an initial guess., $p_0 := 0$ , $x_0 := x_0 / \ x_0\ $ , $\mu_{-1} := (x_0, X_0), w_0 := H x_0 - \mu_{-1} x_0$ do $i=0, 1, \dots$ , until convergence $S_A := \{w_i, x_i, p_i\}^T H \{w_i, x_i, p_i\}$ $S_B := \{w_i, x_i, p_i\}^T \{w_i, x_i, p_i\}$ Solve the smallest eigenvalue $\mu$ and the corresponding vector $v$ , $S_A v = \mu S_B v, v = (\alpha, \beta, \gamma)^T$ . $\mu_i := (\mu + (x_i, H x_i)) / 2$ $x_{i+1} := \alpha w_i + \beta x_i + \gamma p_i, x_{i+1} := x_{i+1} / \ x_{i+1}\ $ $p_{i+1} := \alpha w_i + \gamma p_i, p_{i+1} := p_{i+1} / \ p_{i+1}\ $ $w_{i+1} := T(H x_{i+1} - \mu_i x_{i+1}), w_{i+1} := w_{i+1} / \ w_{i+1}\ $ enddo
--	---

図 2: ランチョス法 (左), と前処理つき共役勾配法 (右).

方向ベクトルを採用し最小固有値を探索する (図 2 右). 基本的に解ベクトル  $x$ , 探索方向ベクトル  $p$ , 残差ベクトル  $w$  のみ保持すればよいことからメモリ要求はランチョス法と変わらないが, 部分空間への射影計算のコストが高いために, これらに予め行列  $H$  を乗じたベクトル (大文字で表記を) 用意する. 解ベクトルもしくは残差ベクトルに応じて探索を打ち切ることができるために, 初期ベクトルの選択や探索方向の良好な選択 (次に述べる前処理も併せて) によって反復回数の制御が行いやすいという利点を有する.

発案者の Knyazev の論文によると, 連立一次方程式と同様に問題行列に応じた前処理を選択することで, その収束特性を劇的に改善できることが報告されている. その意味でも共役勾配法は, 十分に奥深いテーマを含む解法の一つであるといえる. 前処理の選択については使用する計算機のハードウェアや実装にも依存するため別章にて議論する.

### 2.3 実装面からの制約と選択

反復系統の解法の中核部分は行列ベクトル積の実装にある. なぜならば, 行列が疎行列であっても少なくともバンド幅 (行あたりの非ゼロ要素数の最大数) 分のベクトル演算が含まれることになるからである. さらに, 数多くの研究からも明らかなように, 疎行列の行列ベクトル積の並列計算においてデータの格納形式に依存して性能が大きく変化することも知られている.

式 (1) の行列表現は

$$H = I \otimes A + A \otimes I + D, \quad (2)$$

の様にテンソル積を用いた形式 ( $A$  は対称疎行列で,  $D$  は密行列) をしている.  $Hv$  を計算する場合, ベクトル  $v = (v_1, v_2, \dots, v_{N^2})^T$  を並べ直して  $((v_1, \dots, v_N)^T, \dots, (v_{(N-1)N+1}, \dots, v_{N^2})^T)$  を改めて行列  $V$  と表記することで

$$HV = AV + V^T A + D \odot V, \quad (3)$$

と定式化することができる. ここで  $\odot$  は要素毎積演算である.

この表現形式を用いることにより  $Hv$  の演算には自明な並列性とある程度の負荷均衡が保証されていることになる (十分多くの行列ベクトル積を含んでいる). ただし, 分散メモリ計算機上ではデータ分割方式に注意を払わなくてはならず, データの依存関係から自動的に計算順序が決定しさらにデータ転送 (データ再分割) が発生する. 演算に効率的なデータ分散を対応させると,  $AV$  は列分割,  $VA^T$  は行分割であり,

CAL1:  $E^{\text{col}} := \bar{D}^{\text{col}} \odot V^{\text{col}},$   
 CAL2:  $W_1^{\text{col}} := E^{\text{col}} + AV^{\text{col}},$   
 COM1: communication to transpose  $V^{\text{col}}$  into  $V^{\text{row}},$   
 CAL3:  $W_2^{\text{row}} := V^{\text{row}} A^T,$   
 COM2: communication to transpose  $W_2^{\text{row}}$  into  $W_2^{\text{col}},$   
 CAL4:  $W^{\text{col}} := W_1^{\text{col}} + W_2^{\text{col}},$

上の様な流れに従って行列ベクトル積が実行される (上添字の col は列分割, row は行分割を表す). 2 回のデータ転送 COM1, COM2 が必要となるが, 実際の処理では行列データの転置操作となる. 並列処理の用語を用いれば, 全対全の集団通信となるため極めて大きなコストを占めることとなる (隣接プロセッサ間通信ではなく全てのプロセッサが他の全てのプロセッサと送受信する形態). 特別な工夫なしに地球シミュレータ上で行列ベクトル積ルーチンを作成した場合, 上記手順を実行したタイムチャートを図 3 に示す.

図 3 上の NA は 15 億次元ハミルトニアン行列のランチョス法での行列ベクトル積部分に関して, 地球シミュレータ 1 ノード (8 プロセッサ) の処理内容を時間軸に沿って示したものである. 処理中通信と計算が交互に出現しそれらがほぼ同等 (実際は通信が大きい) であるため, システム全体の稼働率は  $1/2$  になる. これは, 如何に地球シミュレータが 40TFLOPS もの高性能を有していても, 20TFLOPS 程度の性能しか出すことができないことを示唆している. この由々しき問題はプログラム実装面からの解決しかありえないが, 適切なアルゴリズム選択も大きく関わってくる.

### 2.3.1 通信と計算のオーバーラップ

図 3 下の TA は地球シミュレータ 1 ノードに搭載されている 8 プロセッサのうち 1 プロセッサを通信処理専用割り当て, 行列ベクトル積を実装した場合のタイムチャートである. 1 プロセッサの計算能力を犠牲にしているため,  $7/8$  の性能上限しか保証されないが, 何の工夫をしなければ  $1/2$  であるので十分許容できるものである. 通信と計算のオーバーラップを施しても, 依然通信のオーバーヘッドが大きいため通信待ちのアイドル状態が多く存在していることがわかる.

### 2.3.2 より積極的な通信隠蔽戦略

行列ベクトル積部分のアイドル状態の解消は, 行列ベクトル積と独立した計算部分を更に通信の裏側で実行することでなされる.

1) ランチョス法は各演算結果の依存性が強いので,  $Hv$  と同時に計算可能な部分が殆ど存在しないためこれ以上の通信隠蔽は期待できない.

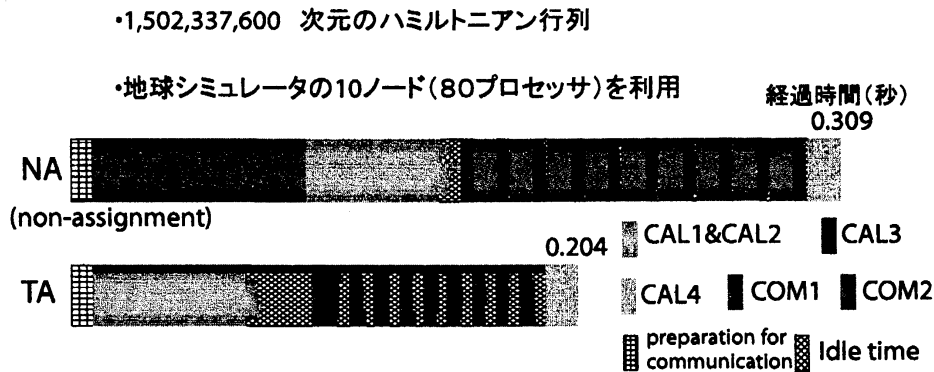


図 3: 行列ベクトル積における通信負荷分布 (NA は特別な工夫なしに実装した場合のもの, TA は通信に従事する専用スレッドを導入して通信と計算のオーバーラップを行ったもの)

2) 一方, 共役勾配法はランチョス法に比べて依存関係が弱く, 数多くの計算を  $Hw$  の背後に持てくることができる (アルゴリズム上は  $Hw$  のみ計算し,  $Hx, Hp$  は別の方法で計算する). 実際,  $(x, p)^T(w, x, p)$  や  $x/\|x\|, p/\|p\|$  など数多くの計算 (開発したプログラムでは最大 11 項目) を  $Hw$  と同時に実行してアイドル状態解消を試みた. しかしながら, 我々が実験した範囲 (各行の非ゼロ要素が 40 程度) では完全に通信を隠蔽することはできず, 依然通信に対して計算量が不足している.

## 2.4 前処理

図 2 は今回の実験において使用したランチョス法と前処理つき共役勾配法の基本アルゴリズムである. 先に示した様に前処理による反復特性の改善が共役勾配法の興味深い点である. アルゴリズム中の  $T$  が前処理に相当する.

前処理も実装依存の部分が大きいのであるが, 前節でクリアすべき問題であった通信の隠蔽は前処理にも関係してくる. 一般に, 前処理は対象となる行列の近似逆行列を乗じることが普通であるため, 反復過程で登場する行列ベクトル積と同様のケアが必要となるのである.

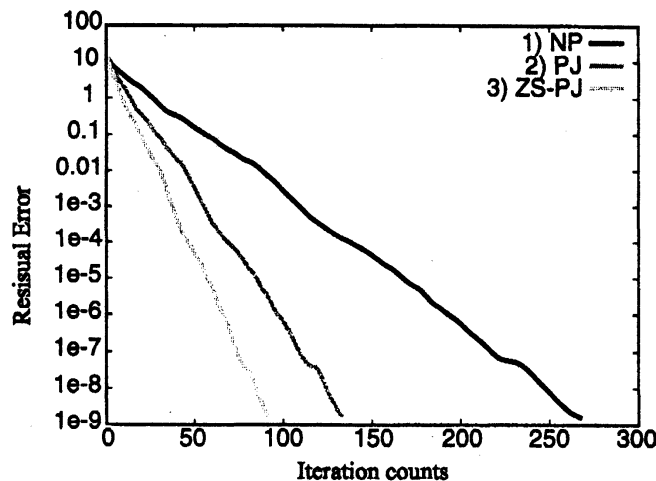
表 1 は,

1. 前処理なし ( $T = I$ )
2. 点ヤコビ ( $T = D^{-1}$ )
3. ゼロシフト付点ヤコビ ( $T = (D - \mu_k)^{-1}$ )

による収束特性の比較を行ったものである (詳細については文献 [7] を参照のこと). よく知られた前処理にはこの他に, ブロックヤコビやノイマン展開近似, SOR などがあるが, これら前処理 (ベースとなる固有値計算アルゴリズムも含めて) を実装する場合に, 行列ベクトル積とほぼ同等の通信コストが必要となるため性能を重視する際は選択が極めて難しくなる. 各種の予備実験を通じて, 全く通信を必要としない上記の前処理の中でもゼロシフト付点ヤコビが総計算時間を改善するという意味で最良の前処理となった.

表 1: 3つの前処理に関する共役勾配法の収束特性の比較.

	1) NP	2) PJ	3) ZS-PJ
反復回数	268	133	91
相対残差 $\ Ax - \mu x\ /\ x\ $	1.445E-9	1.404E-9	1.255E-9
経過時間 (sec)	78.904	40.785	28.205
性能 (FLOPS)	382.55G	383.96G	391.37G



もちろん, SOR などは反復回数を劇的に改善するが [8], 予備実験の結果では行列ベクトル積中に出現するプロセッサ間通信のため 1 反復あたりの処理時間が増加した. 故に, 今回の大規模数値実験では採用しない.

### 3 全固有対計算へのチャレンジ

よりリアルな現象を捉えるために, 2次元格子系での時間発展の計算についてもコード開発を開始している. 時間発展問題を解くためには, 全固有値と固有ベクトルを求める全対角化の必要があり, 最小固有値のみを求める場合とは全く異なるアプローチが必要となる. 一般に疎行列問題に対して, 数個の固有対計算が課される場合には行列を疎行列として扱うことは数値的にも安定であるが (安定な反復法が存在する), 全固有対の計算を行う場合には結局 deflation などを行い直接法的な扱いを施すため, 初めから密行列として扱ったほうがコスト的 (CPU 時間, メモリいずれも) に優れていると判断される.

密対称行列の固有値計算アルゴリズムならびに並列数値計算パッケージとしてよく知られている ScaLAPACK [12] の中から対応するルーチンについてまとめたのが図 4 である. 対称行列をハウスホルダー変換等により三重対角化し, 固有値・固有ベクトルを計算する. 最後に逆変換によりもとの行列の固有ベクトルとする. 固有値・の計算には, 2分法+逆反復法, QR 法, 分割統治法, ツイスト分解法を用いた MRRR 法などが存在するが, 現在の ScaLAPACK に収録されている安定で高速な解法は分割統治法である. テキサス大学オースチンで開発されている PLAPACK [13] の未公開版 (Revision 3.2) に MRRR が収録され

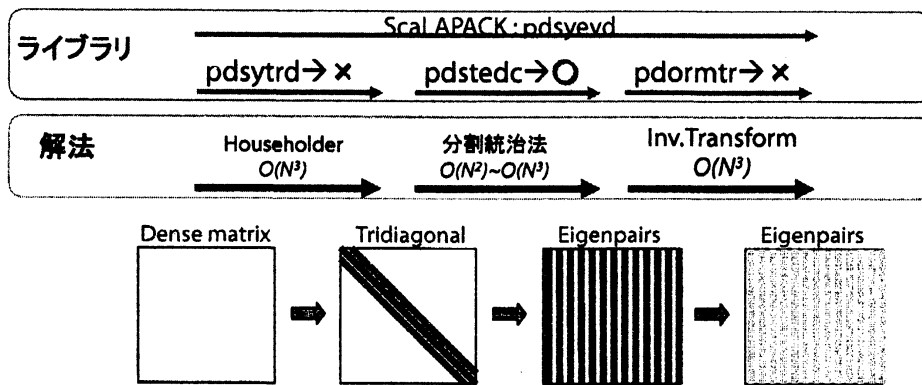


図 4: 密実対称行列の固有値計算の過程

ているが、ハウスホルダー変換の実装に問題があり ScaLAPACK よりも性能面で劣ることが報告されている [14].

我々は、過去に発表された各種の報告から ScaLAPACK のハウスホルダー変換でも性能面で大きな損失があると判断した。実装面でレベル 2BLAS の改良が最重要項目となるが、直野 [9] らの提案するループ融合の手法と Bischof のアルゴリズムを用いた三重対角化ルーチンを開発し [10] その効果を検討した。山本 [11] の報告にあるように、Bischof のアルゴリズムは中間段階の帯行列に変換する過程でレベル 2.5BLAS を利用可能となるため、最大で 2 倍の効果が確認されている。一方で、逆変換のコストが増大するために複数 (特に全固有ベクトル) の場合には不利であると判断される。後述の予備実験 5, 6 では、順変換にいわゆるドンガラのアルゴリズムを採用し、逆変換には WY ブロックアルゴリズムを採用している。実行時間は地球シミュレータでおよそ 7:8, SGI Altix で 3:2 ということで、Bischof を採用するにはドンガラアルゴリズムの順変換部分が高速化されすぎているという知見が得られている (2:1 以上であれば Bischof を採用するメリットがあるのだが...)。さらに、固有値計算のコアアルゴリズムには ScaLAPACK に収録されている分割統治法アルゴリズム (Divide & Conquer 法: pdstedc) を活用する。つまり、図 4 のうち順変換、逆変換は新規開発をし、コア部分は ScaLAPACK の移植作業を行うことになる (なぜなら、地球シミュレータでは ScaLAPACK が正式サポートされていないため)。

現在、開発ルーチンを用いて地球シミュレータと SGI Altix3700Bx2 で予備実験を実施中であり、地球シミュレータの 512CPU (peak 4.096 TFLOPS) を使用して、6 万次元の問題で 1225 秒 (1.652 TFLOPS), 10 万次元の問題で 3907 秒 (2.299 TFLOPS) を記録している。大規模問題での性能向上に向けた改良を実施中であるが、予備実験の一部の結果を図 5 から 7 に示す。SGI Altix 上で開発中のコードは、ベクトル計算機向けの最適化ではなくスカラ計算機向けにキャッシュチューニングしたものであり図 7 が示すように高性能を記録していることが分かる (逆変換ではピークの 70% 強を記録)。

我々の最終ターゲットは 30 万次元以上の系の全対角化である。これは量子素子のシミュレーションであれば約 18 キュービット程度のシミュレーションに相当する。欲を言えばその上を目指したいところである。現在、実用化のための安定動作テストを実施中であり、十分な安定動作確認の後に、近々、本ルーチンを用いた実問題への応用へとシフトする予定



である。

## 4 大規模数値実験結果

基底状態計算の本実験ではランチョス法はメモリ使用量の制約が少ないことから、最大規模の問題を解く際に主解法として利用し、前処理付き共役勾配法をパラメタサーベイ用の主解法とする(ただし本稿では解法の性能比較として可能な限り両解法で計算を実施し掲載した)。

地球シミュレータ(理論最大性能 40TFLOPS)を用いて式(1)のハミルトニアン行列の最小値固有値および、その固有ベクトルを計算した際の結果を以下に示す。ただし、一般利用者が申請可能なパーティションは 512 ノードまでであるので、今回の実験は理論最大性能 32TFLOPS の計算機を利用した結果である。表 2 は、対象となる系の設定と対応するハミルトニアン行列の次元等を示したものである。表からも明らかなように、1000 億次元を超える巨大な数値線形計算であり、要求メモリもテラバイトを超え、国内では地球シミュレータ以外に計算することが不可能な超巨大固有値計算であることが分かる。

続いて、表 3 はハミルトニアン行列の生成から固有値計算に要した時間、更に求められた固有値、固有ベクトルから算出される残差ノルムも併せて記したものである。ランチョス法は反復回数が多く、問題によっては精度が不十分なものも見受けられる。一方、前処理付き共役勾配法では前処理の効果により反復回数が激減しており、性能面においても極めて優秀な成績を残している。モデル 4 では性能 18.86TFLOPS を記録しており、これは我々が行ってきた固有値計算の中で最速のものである。

## 5 まとめ

量子多体系の基底状態計算において、我々が採用した厳密対角化法は使用メモリ量の制約のため大規模系の計算が不可能であったが、今回テラバイト級の計算機利用により従来の記録を大幅に塗り替えることができた。現在このレベルを超える計算機は世界に 2,3 台しかないため、我々の計算は計算物理学において(向こう数年にわたって)資料的価値の高い結果を与えるものである。また、数値計算の視点からも数千億次元の固有値問題を扱うことは極めて稀であり、共役勾配法の利用もこの規模では世界初ではないだろうか。また、計算機科学の観点からも通信が支配的な並列プログラムにおいてシステムの 50% 以上を超える性能をたたき出した点は学会からも評価されたように特筆すべき点と考えている。

また、全対角化についても世界最大規模の計算に関する準備段階にあるが、そこにおける現時点での成果を示した。すでに 20 万次元の完全対角化を実現しており、実用の目前まで来ている。さらに、性能面では ScaLAPACK の三重対角化ルーチンを大きく凌駕する性能を記録しており、十分に満足いくものが完成している。今後はこれら対角化ルーチン群の安定化とともに、実アプリケーションへの適用が重要であろう。それに向けたプログラムのシフト体制も着々と進行中であり、近日中に成果を報告できると確信している。

最後に、計算機実験にあたり海洋研究開発機構・地球シミュレータセンターの支援を得た。また、研究遂行にあたり日本原子力研究開発機構システム計算科学センターの矢川元基センター長、平山俊雄次長からは多くの助言を頂いた。関係各所に感謝の意を表したい。

1152CPUs(Front), 576CPUs(Back) on the ES.

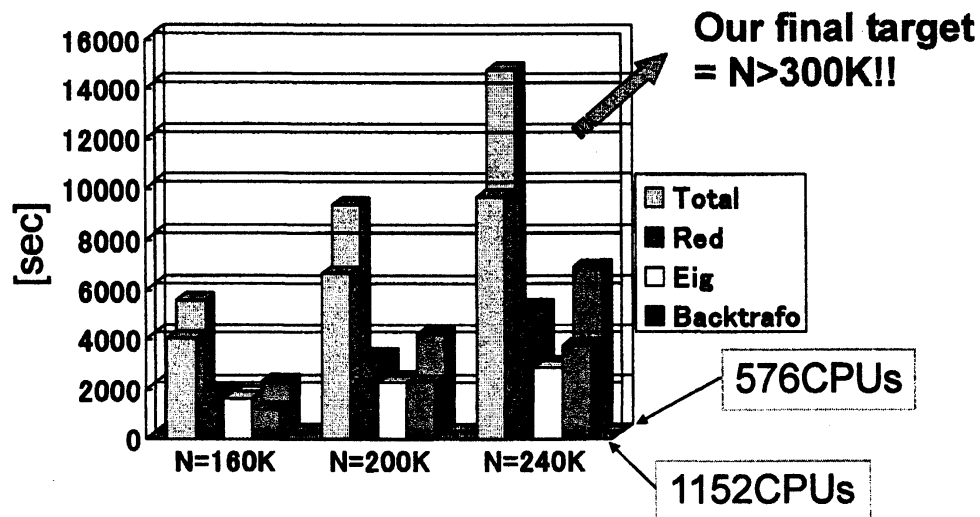


図 5: 地球シミュレータでの実対称行列の全固有対の計算時間内訳 (Total: 全計算時間, Red[uction]:三重対角化, Eig[en]: 分割統治法, Backtrafo:逆変換). 前方グラフが 1152CPU 使用, 後方グラフが 576CPU 使用での測定結果.

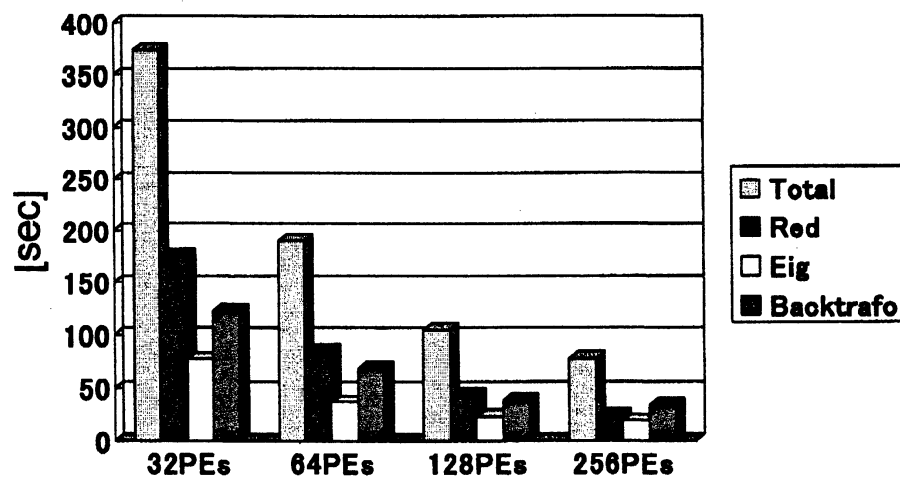


図 6: SGI Altix3700Bx2 での実対称行列 ( $N = 20,000$ ) の全固有対の計算時間内訳 (Total: 全計算時間, Red[uction]:三重対角化, Eig[en]: 分割統治法, Backtrafo:逆変換)

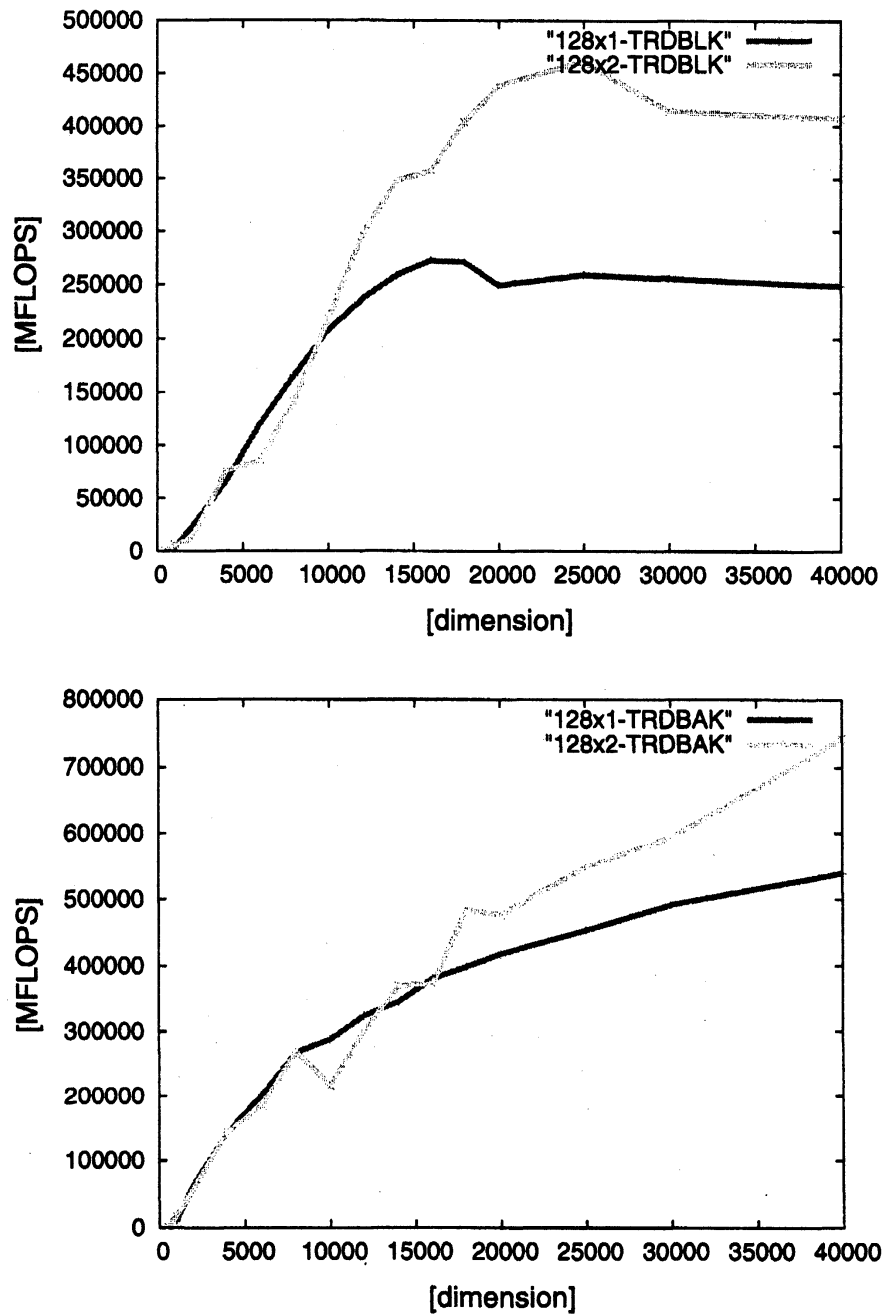


図 7: SGI Altix3700Bx2 での三重対角化ルーチンの性能 (上: 順変換, 下: 逆変換). ハードウェアの理論ピーク性能は 128CPU 使用時 819GFLOPS, 256CPU で 1638GFLOPS.

## 参考文献

- [1] M.Machida, S.Yamada, Y.Ohashi, H.Matsumoto: "Novel Superfluidity in a Trapped Gas of Fermi Atoms with Repulsive Interaction Loaded on an Optical Lattice", *Phys. Rev. Lett.*, **93** (2004) 200402
- [2] E.Dagotto, "Correlated electrons in high-temperature superconductors", *Rev. Mod. Phys.*, Vol. 66, pp.763, 1994.
- [3] 地球シミュレータセンター. <http://www.es.jamstec.go.jp/>
- [4] 山田進, 町田昌彦, 今村俊幸, 強相関電子系における超大規模固有値問題-地球シミュレータでのベクトル並列計算-, 情報処理学会論文誌 コンピューティングシステム, Vol. 45, No. SIG6 (ACS6), pp.161-170, 2004.
- [5] S.Yamada, T.Imamura, M.Machida: "16.447 TFlops and 159-Billion-dimensional Exact-diagonalization for Trapped Fermion-Hubbard Model on the Earth Simulator", In *Proc. of Supercomputing 2005 (SC05)*, Seattle, 2005.
- [6] A.V.Knyazev: "Preconditioned eigensolvers - An oxymoron?", *Electronic Trans. on Numer. Anal.*, Vol. 7, pp.104-123, 1998.
- [7] 山田進, 今村俊幸, 町田昌彦, 前処理付き共役勾配法による超大規模行列の固有値計算, 2005年度日本応用数学会年會講演集, pp.190-191, 2005.
- [8] 山田進, 今村俊幸, 町田昌彦, 量子固有値問題に対する前処理付き共役勾配法: 適応的シフト前処理の収束性, 日本応用数学会 平成 18 年研究部会 連合発表会, 2006.
- [9] K.Naono, Y.Yamamoto, M.Igai, and H.Hirayama: "High performance implementation of tridiagonalization on the SR8000", In *Proc. of the High Performance Computing in Asia-Pacific Region (HPC-ASIA2000)*, Beijing, pp.206-219, 2000.
- [10] C.Bischof, X.Sun, and B.Lang, "Parallel tridiagonalization through two-step band reduction", In *Proc. of Scalable High Performance Computing Conference*, pp.23-27. IEEE, 1994.
- [11] 山本有作, キャッシュマシン向け対称密行列固有値解法の性能・精度評価, 情報処理学会論文誌 コンピューティングシステム, Vol. 46, No. SIG3 (ACS8), pp. 81-91, 2005.
- [12] ScaLAPACK ホームページ, [http://www.netlib.org/scalapack/scalapack\\_home.html](http://www.netlib.org/scalapack/scalapack_home.html)
- [13] P.Alpatov, et al.: "PLAPACK: Parallel Linear Algebra Package," in *Proc. of the SIAM Parallel Processing Conference*, 1997 <http://www.cs.utexas.edu/~plapack/>
- [14] E.Breitmoser, A.G.Sunderland, "A performance study of the PLAPACK and ScaLAPACK Eigensolvers on HPCx for the standard problem", Technical Report from the HPCx Consortium, 2004. <http://www.hpcx.ac.uk/research/hpc/HPCxTR0406.pdf>

表 2: ハミルトニアン行列  $H$  の次元, 地球シミュレータを使用した際のノード数, メモリ使用量など. ただし, モデル 1~3 は 1 次元格子, モデル 4 は  $6 \times 4$  の 2 次元格子の 2 角を除いた 22 格子から構成される系.

Model	No. of Sites	Fermions ( $\uparrow / \downarrow$ spin)	Dimension of $H$	No. of Nodes	Memory (TB)	
					Lanczos	PCG
1	24	7 / 7	119,787,978,816	512	5.4	na
2	22	8 / 8	102,252,852,900	512	4.6	6.9
3	22	9 / 8	159,059,993,400	512	7.1	na
4	22*	8 / 8	102,252,852,900	512	4.6	6.9

表 3: 地球シミュレータでのランチョス法と前処理付き共役勾配法の実効性能 (上表: ランチョス法, 下表: 前処理付き共役勾配法).

Model	Itr.	Residual Error	Elapsed time(sec)	
			Total	Solver
1 (TFLOPS)	300	$1.3 \times 10^{-11}$	275.368 (12.737)	266.729 (13.116)
2 (TFLOPS)	300	$3.6 \times 10^{-11}$	288.270 (10.613)	279.775 (10.906)
3 (TFLOPS)	300	$4.2 \times 10^{-8}$	362.635 (13.363)	352.944 (13.691)
4 (TFLOPS)	—	—	—	—

Model	Itr.	Residual Error	Elapsed time (sec)	
			Total	Solver
1 (TFLOPS)	—	—	—	—
2 (TFLOPS)	109	$2.4 \times 10^{-9}$	66.819 (14.763)	59.510 (16.447)
3 (TFLOPS)	—	—	—	—
4 (TFLOPS)	29	$8.6 \times 10^{-9}$	25.7290 (13.957)	18.9597 (18.868)